

EXHIBIT 5

Filed Under Seal

ATTORNEYS' EYES ONLY – SOURCE CODE

- A first Accused Google Controller comprising a Google Pixel 7 device running Android version 13 that was installed with the Google Home (version 2.60.60.4), YouTube Music (version 5.33.51), and Spotify (version 8.7.84.382) apps for Android, which I will refer to below as the “Android-based Google Controller”
- A second Accused Google Controller comprising Google Pixelbook device running ChromeOS version 107.0.5304.110 that was installed with the Google Home (version 2.60.1.19), YouTube Music (version 5.32.50), and Spotify apps for ChromeOS, which I will refer to below as the “ChromeOS-based Google Controller”
- A third Accused Google Controller comprising an iPhone 12 Pro device running iOS version 16.1.1 that was installed with the Google Home (version 2.60.116), YouTube Music (version 5.13.1), and Spotify (version 8.7.82.107) apps for iOS, which I will refer to below as the “iOS-based Google Controller” (this was the same iPhone 12 Pro that was used in connection with the testing I described in my June 22, 2022 Opening Report for the “Patent Showdown”)
- A first Accused Google Player comprising a Nest Hub player running Cast firmware version 1.56.309385, which was named “Kitchen” (this was the same Nest Hub player that was used in connection with the testing I described in my June 22, 2022 Opening Report for the “Patent Showdown”)
- A second Accused Google Player comprising a Home Mini player running Cast firmware version 1.56.313652, which was named “Master Bedroom” (this was the same Home Mini player that was used in connection with the testing I described in my June 22, 2022 Opening Report for the “Patent Showdown”)
- A third Accused Google Player comprising a Nest Audio player running Cast firmware version 1.56.313652, which was named “Living Room” (this was the same Nest Audio player that was used in connection with the testing I described in my June 22, 2022 Opening Report for the “Patent Showdown”)

42. Almeroth Op. Rep. ¶193. However, I understand that Sonos’s test devices (*e.g.*, Google’s Nest Hub, Nest Audio, Nest Mini, Pixel, and Apple’s iPhone, Almeroth Op. Rep. ¶146) are running firmware version 1.56.324896, which is later than the version that Dr. Almeroth tested. Ex. 3, (2022-12-13 Email fr Richter to Kaplan).

43. Furthermore, Dr. Almeroth did recognize that Google produced the new version of source code corresponding to firmware version 1.56.324896 but did not review or analyze that source code. Almeroth Op. Rep. p. 23 n.3 (“Thus, it appears that none of the representative Accused Google Players or the representative Accused Google Controllers in the Google test system had been updated with the “new version of [Google’s] source code.” However, I understand that recently Google allegedly produced this “new version of its source code,” but I

ATTORNEYS' EYES ONLY – SOURCE CODE

have not yet had a chance to inspect the “new” code. Thus, I reserve my right to address the “new” code in a supplement to this Report and/or in my reply Report after I have inspected the “new” code and/or have been provided with additional evidence related to the “new” code.”). I understand that Sonos’s attorneys have reviewed this source code, and that the relevant source code was produced by Google in August. See Ex. 4 (2022-08-25 Email fr Leroy to Richter).

44. Accordingly, although Dr. Almeroth allegedly offers opinions about the “current” functionality of Google’s Accused Products, that is not accurate and the products that Dr. Almeroth tested and opines upon in his Report do not contain the latest firmware updates for those products.

B. Grouping Functionality

45. Users can create or modify speaker groups for Google speakers using the Google Home app. In particular, as shown below, users can use the “+” item in the Google Home app to “create speaker group,” and from there a user may select speakers to add to that speaker group and subsequently provide a name for the speaker group.

ATTORNEYS' EYES ONLY – SOURCE CODE

59. The foregoing is confirmed by my review of the relevant source code running on Google speakers. For example, I reviewed source code within the multizone_manager.cc file. In the snippet below, the code on a speaker iterates through a list of all the potential existing groups that the speaker could join; the speaker checks to see if it is already a member of the group it is being asked to join. If it is not, it stops whatever it is currently playing and begins the process to add itself to that group. If it is already a member of the group it is being asked to join, then it saves any updated information about the group and otherwise retains its current behavior. I note that this code is found in the MultizoneManager::RefreshDeviceGroups() function definition and applies to the process of refreshing device groups. The functionality of this source code is also consistent with my testing of the functionality of the accused players, which I set forth above. In the case where a new group is created, the players within that group cease to function as individual players and are instead stopped and are being controlled as a group. In the case where an existing group is modified by adding a new device to that group, the new device becomes part of the reconfigured group, and immediately adopts the configuration of the group. Therefore, in neither case do players added to a new or existing group continue to operate in an individual playback mode that the speakers were in prior to being added to the group. In addition, any leader election for the group leader would take place after the AddGroup(g) command.

```
base::flat_set<std::string> group_uuids({virtual_group_uuid_});
for (const auto& g : local_groups) {
    group_uuids.insert(g.uuid);
    auto it = groups_.find(g.uuid);
    if (it == groups_.end()) {
        StopCurrentApp();
        AddGroup(g);
    } else if ((it->second->Reconfigure(g)) {
        SaveGroupConfig(g);
    } else {
        continue;
    }
}
```

ATTORNEYS' EYES ONLY – SOURCE CODE

```
}
```

```
groups_changed = true;
```

```
SC-GOOG-SONOSNDCA-001637-38.
```

X. NO EVIDENCE OF INFRINGEMENT

A. No Evidence of Indirect Infringement

60. I understand that a patent may be infringed indirectly. I understand that two types of indirect infringement are induced infringement and contributory infringement. I further understand that indirect infringement requires a threshold showing of direct infringement. In my opinion, Dr. Almeroth has failed to show that Google indirectly infringes any asserted claim.

61. For each asserted patent and accused “product,” Dr. Almeroth provides essentially the same opinion regarding indirect infringement. Dr. Almeroth asserts that Google induces infringement of the asserted patents by promoting and encouraging the infringing use of the Accused Products. For each, Dr. Almeroth generally points to marketing materials regarding the Accused Products. He appears to base his conclusion on these general statements, with no attempt to tie them to specific elements that Dr. Almeroth points to in his purported element-by-element analysis. To the extent Dr. Almeroth is permitted to submit this opinion, he does not actually show any attempt to encourage *infringing use* at all.

62. This is true especially because the scope of the claimed invention is narrowly prescribed. Merely playing music on a speaker alone does not infringe. Merely creating a speaker group alone does not infringe. Merely playing music to a speaker group alone does not infringe. Merely naming a speaker group alone does not infringe. The situation where a user asks one of his or her speakers a question or provides an instruction to the speakers does not infringe. A user casting a video or movie to his or her TV does not by itself infringe. There are